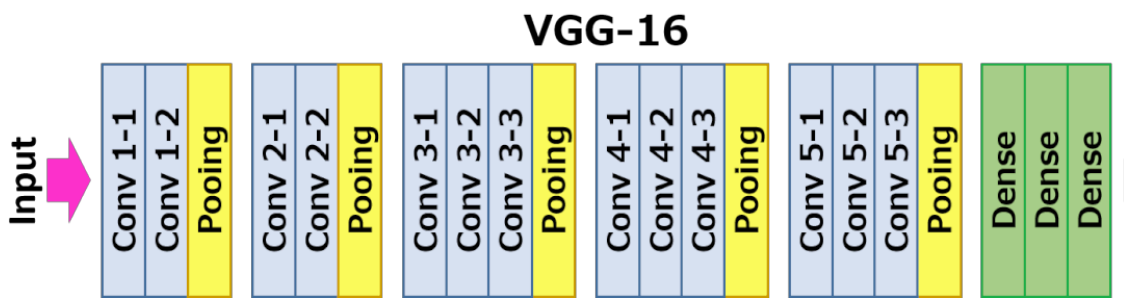
Harman Singh Chawla
Net ID Hsc367

Sahil Goel
NetID sg5591

Architecture used: VGG16



Architecture design:
VGG16 -> output flattened -> dropout -> dense layer (output) with sigmoid activation

Duration of training: 30 epochs (1486.9s)

Augmentations performed: Rotation, Vertical and Horizontal flip

Input Size: Number of images in the training data stay the same for each epoch. We swap and add augmented data randomly through the epochs instead of training it once on all images and then on augmented data.

Accuracy on training: ~ 73%

Observations:
1. In my experience, I found it better to play with the data rather than the model architecture itself to increase the accuracy and reduce the result. The data seems to be respond well to gaussian smoothing and cropping. The model we used takes a 212*212 as the input. The larger image input allows the network to access more data values to refine its parameters and seemed to work way bette than the reduced 106*106*3 image.
2. The architecture in place (VGG16) is a model sufficient enough to handle Galaxy Zoo 2 data. I didn't see a need to go deeper although I did try ResNet50 and VGG19.
3. The augmentations to the data need to be chosen wisely as sheering the image or extreme rotations lead to overlaps and thus a lower accuracy.
4. I used sigmoid as the activation function for the last dense layer as softmax returns the probability distribution and in our case, the probability of a galaxy being of one class didn't seem to affect the probability of another.
5. I used the Adam optimizer instead of RMSProp defined in baseline. Now, to lower the learning rate as the training progressed and I have used a callback ReduceLR from Keras defined on the loss observed.
6. Some other things I wanted to try was performing region segmentation (through edge detection) on the cropped image and then making all pixels which were not the galaxy black. This would have removed any background noise disturbing the result.